A note on characterizations of context-free languages using insertion and locality (Theoretical Computer Science and Its Applications)

AUTHOR(S):

Onodera, Kaoru

数理解析研究所講究録
第1649巻 2009年 23-30

2008年度冬のLAシンポジウム [5]

A note on characterizations of context-free languages using insertion and locality

小野寺 薫 (Kaoru Onodera)

東京電機大学 理工学部 サイエンス学系
Division of Sciences, School of Science and Engineering, Tokyo Denki University,
Ishizaka, Hatoyama-machi, Hiki-gun, Saitama 350-0394, JAPAN,
kaoru@j.dendai.ac.jp

Abstract. In this paper, we obtain some characterizations and representation theorems of context-free languages in Chomsky hierarchy by using insertion systems, strictly locally testable languages, and morphisms. For instance, each context-free language L can be represented in the form $L = h(L(\gamma) \cap R)$, where γ is an insertion system of weight $(3, 0)$, R is a strictly 3-testable language, and h is a projection.

1 Introduction

DNA computing theory involves the use of *insertion* and *deletion operations*. Insertion systems in which we can use only insertion operations are somewhat intermediate between Chomsky context-sensitive grammars and Marcus contextual grammars. From the definition of insertion operations, if there is no context-checking to insertion operation, one would imagine that by using only insertion operations, we generate only context-free languages.

On the other hand, the class of strictly locally testable languages is known as a proper subclass of regular language classes [1]. The equivalence relation between a certain type of splicing languages and strictly locally testable languages is known [2].

The well-known Chomsky-Schützenberger representation theorem implies that any context-free language is a homomorphic image of a Dyck language and a regular language. An analogous representation was considered in [3], which shows that any context-free language can be expressed in the form $h(L(\gamma) \cap R)$, where γ is an insertion system, h is a projection, and R is a star language.

In this paper, we focus on characterizing context-free languages by using insertion systems together with strictly locally testable languages and morphisms.

In insertion systems, a pair of the maximum length of inserted strings and the one of context-checking strings, called *weight* is an important parameter for generative powers. As for strictly locally testable languages, the length of local testability-checking is considered.

We prove that each context-free language can be represented in the form $h(L(\gamma) \cap R)$, where γ can be simplified to be of weight $(2,0)$, h is a morphism, and R is a strictly 3-testable language.

2 Preliminaries

In this section, we introduce necessary notation and basic definitions needed in this paper. We assume the reader to be familiar with the rudiments on basic notions in formal language theory (see, e.g., [4, 5]).

2.1 Basic Definitions

For an alphabet V, V^* is the set of all strings of symbols from V which includes the empty string λ. For a string $x \in V^*$, $|x|$ denotes the length of x. For $0 \leq k \leq |x|$, let $Pre_k(x)$ and $Suf_k(x)$ be the prefix and the suffix of x of length k, respectively. For $0 \leq k \leq |x|$, let $Int_k(x)$ be the set of *proper* interior substrings of x of length k, while if $|x| = k$ then $Int_k(x) = \emptyset$.

2.2 Normal Forms of Grammars

A *phrase structure grammar* is a quadruple $G = (N, T, P, S)$, where N is a set of *nonterminal symbols*, T is a set of *terminal symbols*, P is a set of *production rules*, and S in N is the *initial symbol*. A rule in P is of the form $r : \alpha \to \beta$, where $\alpha \in (N \cup T)^* N (N \cup T)^*$, $\beta \in (N \cup T)^*$, and r is a label from a given set $Lab(P)$ such that there are no production rules with the same label. For any x and y in $(N \cup T)^*$, if $x = u\alpha v$, $y = u\beta v$, and $r : \alpha \to \beta \in P$, then we write $x \overset{r}{\Longrightarrow}_G y$. We say that x *directly derives* y with respect to G. If there is no confusion, we write $x \Longrightarrow y$. The reflexive and transitive closure of $\Longrightarrow$ is denoted by $\Longrightarrow^*$.

We define a *language* $L(G)$ generated by a grammar G as follows:

$$L(G) = \{w \in T^* \mid S \Longrightarrow_G^* w\}.$$

It is well known that the class of languages generated by the phrase structure grammars is equal to the class of *recursively enumerable languages* RE [5].

A grammar $G = (N, T, P, S)$ is *context-free* if P is a finite set of *context-free rules* of the form $A \to \alpha$, where $A \in N$ and $\alpha \in (N \cup T)^*$. A language L is a *context-free language* if there is a context-free grammar G such that $L = L(G)$. Let CF be the class of context-free languages.

A context-free grammar $G = (N, T, P, S)$ is in *Chomsky normal form* if each production rule in P is of one of the following forms:

1. $X \to YZ$, where $X, Y, Z \in N$.

2. $X \to a$, where $X \in N$, $a \in T$.
3. $S \to \lambda$ (only if S does not appear in right-hand sides of production rules).

It is well known that, for each context-free language L, there is a context-free grammar in Chomsky normal form generating L [5].

A grammar $G = (N, T, P, S)$ is *regular* if P is a finite set of rules of the form $X \to \alpha$, where $X \in N$ and $\alpha \in TN \cup T \cup \{\lambda\}$. A language L is a *regular language* if there is a regular grammar G such that $L = L(G)$. Let REG be the class of regular languages.

We are going to define a strictly locally testable language, which is one of the main objectives of the present work.

Let k be a positive integer. A language L over T is strictly k-testable if there is a triplet $S_k = (A, B, C)$ with $A, B, C \subseteq T^k$ such that, for any w with $|w| \geq k$, w is in L iff $Pre_k(w) \in A$, $Suf_k(w) \in B$, $Int_k(w) \subseteq C$.

Note that if L is strictly k-testable, then L is strictly k'-testable for all $k' > k$. Further, the definition of strictly k-testable says nothing about the strings of "length $k-1$ or less".

A language L is *strictly locally testable* iff there exists an integer $k \geq 1$ such that L is strictly k-testable. Let $LOC(k)$ be the class of strictly k-testable languages. Then one can prove the following theorem.

Theorem 1. [6] $LOC(1) \subset LOC(2) \subset \cdots \subset LOC(k) \subset \cdots \subset REG$.

We are now going to define an insertion system. An *insertion system* is a triple $\gamma = (T, P, A_X)$, where T is an alphabet, P is a finite set of *insertion rules* of the form (u, x, v) with $u, x, v \in T^*$, and A_X is a finite set of strings over T called axioms.

We write $\alpha \overset{r}{\Longrightarrow}_\gamma \beta$ if $\alpha = \alpha_1 uv\alpha_2$ and $\beta = \alpha_1 uxv\alpha_2$ for some insertion rule $r : (u, x, v) \in P$ with $\alpha_1, \alpha_2 \in T^*$. If there is no confusion, we write $\alpha \Longrightarrow \beta$. The reflexive and transitive closure of $\Longrightarrow$ is defined by $\Longrightarrow^*$.

A language generated by γ is defined by

$$L(\gamma) = \{w \in T^* \mid s \Longrightarrow_\gamma^* w, \text{ for some } s \in A_X\}.$$

An insertion system $\gamma = (T, P, A_X)$ is said to be of *weight* (m, n) if

$$\begin{aligned} m &= max\{\ |x| \mid (u, x, v) \in P\}, \\ n &= max\{\ |u| \mid (u, x, v) \in P \text{ or } (v, x, u) \in P\}. \end{aligned}$$

For $m, n \geq 0$, INS_m^n denotes the class of all languages generated by insertion systems of weight (m', n') with $m' \leq m$ and $n' \leq n$. When the parameter is not bounded, we replace m or n with $*$.

For insertion systems, there exist the following results.

Theorem 2. [4]

1. *For any* $0 \geq m \geq m'$ *and* $0 \geq n \geq n'$, $INS_m^n \subseteq INS_{m'}^{n'}$.
2. $FIN \subset INS_*^0 \subset INS_*^1 \cdots \subset INS_*^* \subset CS$.
3. $REG \subset INS_*^*$.
4. $INS_*^1 \subset CF$.
5. *CF is incomparable with all* INS_*^n $(n \geq 2)$, *and* INS_*^*.
6. INS_2^2 *contains non-semilinear languages.*

A mapping $h : V^* \to T^*$ is called *morphism* if $h(\lambda) = \lambda$ and $h(xy) = h(x)h(y)$ for any $x, y \in V^*$. For languages L_1, L_2, and morphism h, we use the following notation: $h(L_1 \cap L_2) = \{h(w) \mid w \in L_1 \cap L_2\}$. For language classes $\mathcal{L}_1$ and $\mathcal{L}_2$, we introduce the following class of languages:

$$H(\mathcal{L}_1 \cap \mathcal{L}_2) = \{h(L_1 \cap L_2) \mid h \text{ is a morphism}, L_i \in \mathcal{L}_i \ (i = 1, 2)\}.$$

3 Characterizations of Context-Free Languages

We will show how context-free languages can be characterized by insertion systems and strictly locally testable languages.

3.1 Characterizations developed from Păun's result

In some respect the proof technique for $CF = H(INS_3^0 \cap R_S)$, where R_S is the class of star languages [3], might be helpful to follow the proof of this subsection.

Lemma 1. $CF \subseteq H(INS_3^0 \cap LOC(4))$.

Proof. Consider a context-free grammar $G = (N, T, P, S)$ in Chomsky normal form. We construct an insertion system $\gamma = (\Sigma, P_\gamma, \{S\})$, where

$$\begin{aligned}
\Sigma &= V \cup \bar{V} \cup T, \\
V &= N \cup Lab(P), \\
P_\gamma &= \{(\lambda, YZr, \lambda), (\lambda, \bar{X}\bar{r}, \lambda) \mid r : X \to YZ \in P\} \cup \\
&\quad \{(\lambda, ar, \lambda), (\lambda, \bar{X}\bar{r}, \lambda) \mid r : X \to a \in P\} \cup \\
&\quad \{(\lambda, r, \lambda), (\lambda, \bar{S}\bar{r}, \lambda) \mid r : S \to \lambda \in P\}.
\end{aligned}$$

For the rule $r : X \to \alpha$ in P, we say that the two insertion rules $(\lambda, \alpha r, \lambda)$ and $(\lambda, \bar{X}\bar{r}, \lambda)$ in P_γ are *r-pair.*

We define the projection $h : \Sigma^* \to T^*$ by

$$\begin{aligned}
h(a) &= a \quad \text{for all } a \in T, \\
h(a) &= \lambda \quad \text{otherwise.}
\end{aligned}$$

Consider $R = A\Sigma^* \cap \Sigma^* B - \Sigma^+ C' \Sigma^+$ with $C' = \Sigma^4 - C$, where

$$A = \{arX\bar{X} \mid r : X \to a \in P\} \cup \{rS\bar{S}\bar{r} \mid r : S \to \lambda \in P\},$$
$$B = \{rS\bar{S}\bar{r} \mid r : S \to \alpha \in P,\ \alpha \in (N \cup T)^*\},$$
$$C = \{rX\bar{X}\bar{r},\ X\bar{X}\bar{r}a,\ X\bar{X}\bar{r}r_1,\ \bar{X}\bar{r}ar_1,\ \bar{X}\bar{r}r_1Y,\ \bar{r_1}arX,\ \bar{r_1}rX\bar{X},\ arX\bar{X} \mid$$
$$r : X \to \alpha \in P,\ r_1 : Y \to \alpha_1 \in P,\ a \in T,\ \alpha, \alpha_1 \in (N \cup T)^*\}.$$

Then R is a strictly 4-testable language prescribed by $S_4 = (A, B, C)$. The language R can be characterized by using

$$\Omega = \{rX\bar{X}\bar{r} \mid r : X \to \alpha \in P,\ \alpha \in (N \cup T)^*\}$$

such that $R \subset (\Omega \cup T\Omega)^*(B \cup TB)$. A nonterminal symbol X in $rX\bar{X}\bar{r} \in \Omega$ is said to be *Ω-blocked*. A symbol in $N \cup T$ which is not Ω-blocked is said to be *unblocked*. Intuitively, an Ω-blocked nonterminal symbol X in $rX\bar{X}\bar{r}$ means that X has been used for the rule r.

Further, based on γ and R, we define the followings: for each $X \in N$, let

$$\gamma_X = (\Sigma, P_\gamma, \{X\})$$

be an insertion grammar, and let

$$R_X = A\Sigma^* \cap \Sigma^* B_X - \Sigma^+ C' \Sigma^*$$

be a strictly 4-testable language, where $B_X = \{rX\bar{X}\bar{r} \mid r : X \to \alpha \in P,\ \alpha \in (N \cup T)^*\}$. There is a slight note on the form of $\Sigma^+ C' \Sigma^*$ in R_X. Then R_X can be characterized by $R_X \subset (\Omega \cup T\Omega)^*$. For the case $X = S$, $\gamma_S = \gamma$ and $B_S = B$ hold.

We can prove that, for any X in N, if there is a derivation $X \overset{r_1 \cdots r_n}{\Longrightarrow}_G a_1 \cdots a_l$ with $a_i \in T$ $(1 \le i \le l)$ then there is a string

- $w = a_1 u_1 \cdots a_l u_l$ in $L(\gamma_X) \cap R_X$,
  where $l \ge 2$, $u_i \in \Omega^+$ $(1 \le i \le l-1)$, and $u_l \in \Omega^*\{r_1 X\bar{X}\bar{r}_1\}$, or
- $w = a_1 u_1$ in $L(\gamma_X) \cap R_X$,
  where $u_1 \in \Omega^*\{r_1 X\bar{X}\bar{r}_1\}$

by induction on the length n of derivations in G. We omit the proof here.

Conversely, we will show that, for a string w in $L(\gamma) \cap R$, $h(w)$ is in $L(G)$, which can be derived from showing that if a string w is in $L(\gamma_X) \cap R_X$, then there is a derivation $X \Longrightarrow_G^* h(w)$. We omit the proof here. □

It is known that the class of context-free languages includes the class of insertion languages of weight $(3, 0)$ [7]. Together with the fact that the class of context-free languages is closed under intersection with regular languages and morphisms, we have $H(INS_3^0 \cap LOC(4)) \subseteq CF$, which indicates the following theorem.

Theorem 3. $CF = H(INS_3^0 \cap LOC(4))$.

Furthermore, from Theorem 1, we have the following corollary.

Corollary 1. $CF = H(INS_3^0 \cap LOC(k))$ ($k \geq 4$).

3.2 Characterization developed from Chomsky-Schützenberger representation

For an alphabet Σ, let $\bar{\Sigma} = \{\bar{x} \mid x \in X\}$ be a barred copy of Σ. Σ and $\bar{\Sigma}$ are considered to be disjoint. Then *Dyck language* D over Σ and $\bar{\Sigma}$ is defined to be the context-free language generated by the grammar $G_D = (\{S\}, \Sigma \cup \bar{\Sigma}, P, S)$, where $P = \{S \to SS,\ S \to aS\bar{a},\ S \to \epsilon \mid a \in \Sigma, \bar{a} \in \bar{\Sigma}\}$. Let *Dyck* be a class of Dyck languages.

To show the equality $CF = H(INS_2^0 \cap LOC(3))$, we first consider the following theorem.

Theorem 4. $H(Dyck \cap REG) \subseteq H(Dyck \cap LOC(3))$.

Proof. Let $h_1 : T^* \to \Gamma^*$ be a morphism, D be Dyck language over $\Sigma \cup \bar{\Sigma}$, and $G = (N, T, P, S)$ with $T = \Sigma \cup \bar{\Sigma}$ be a regular grammar. We construct Dyck language D', strictly 3-testable languages, and morphism h_2 as follows.

- Strictly 3-testable languages.
  For any $N_1, N_2 \in N$, we construct

$$L(N_1 : N_2) = A(N_1)\Sigma^* \cap \Sigma^* B(N_2) - \Sigma^+ C' \Sigma^+$$

with $C' = \Sigma^3 - C$, where

$$\begin{aligned} A(N_1) &= \{N_1\bar{N}_1 a \mid N_1 \to aX \in P,\ X \in N,\ a \in T\}, \\ B(N_2) &= \{aN_2\bar{N}_2 \mid X \to aN_2 \in P,\ X \in N, a \in T\}, \\ C &= \{\bar{X}aY,\ aY\bar{Y},\ X\bar{X}a \mid X \to aY \in P,\ X, Y \in N,\ a \in T\}. \end{aligned}$$

By using the new symbols F and $\bar{F}$, for any $N_1 \in N$, we construct

$$L(N_1 : F) = A(N_1 : F)\Sigma^* \cap \Sigma^* B(N_1 : F) - \Sigma^+ C'(F) \Sigma^+$$

with $C' = \Sigma^3 - C$, where

$$\begin{aligned} A(N_1 : F) = {} & \{N_1\bar{N}_1 a \mid N_1 \to aX \in P \text{ or } N_1 \to a \in P,\ a \in T,\ X \in N\} \cup \\ & \{S\bar{S}F \mid N_1 = S,\ S \to \lambda \in P\}, \\ B(N_1 : F) = {} & \{aF\bar{F} \mid X \to a \in P,\ a \in T\} \cup \\ & \{SF\bar{F} \mid N_1 = S,\ S \to \lambda \in P\}, \\ C(F) = {} & \{\bar{X}aY,\ aY\bar{Y},\ X\bar{X}a \mid X \to aY \in P,\ X, Y \in N,\ a \in T\} \cup \\ & \{X\bar{X}a,\ \bar{X}aF \mid X \to a \in P,\ X \in N, a \in T\}. \end{aligned}$$

From the above definitions, for any $N_1, N_2 \in N$, the followings hold:

$$A(N_1), B(N_2) \subset C, \quad A(N_1) \subseteq A(N_1 : F), \quad C \subset C(F).$$

- Dyck language D'.
  By using the new symbols F and $\bar{F}$, D' is a Dyck language over $\Sigma \cup N \cup \{F\}$ and $\bar{\Sigma} \cup \bar{N} \cup \{\bar{F}\}$.
- Homomorphism h_2.
  For $V = T \cup N \cup \bar{N} \cup \{F, \bar{F}\}$, we define $h_2 : V^* \to \Gamma^*$ by

$$\begin{aligned} h_2(a) &= h_1(a) \quad a \in T, \\ h_2(a) &= \epsilon \quad \text{otherwise.} \end{aligned}$$

We will prove that $h_1(D \cap L(G)) = h_2(D' \cap L(S : F))$.
$[h_1(D \cap L(G)) \subseteq h_2(D' \cap L(S : F))]$

To show the inclusion, we first prove that for any x which satisfies that $x \in D$, $|x| = 2n$, and $X \Longrightarrow_G^* xY$ with $X, Y \in N$, there is a string $y \in D' \cap L(X : Y)$ such that $h_2(y) = h_1(x)$ by induction on n. We omit the proof here. The inclusion $h_1(D \cap L(G)) \subseteq h_2(D' \cap L(S : F))$ can be proved by considering the case $X = S$ and $Y = F$ in the previous claim.
$[h_1(D \cap L(G)) \supseteq h_2(D' \cap L(S : F))]$

We will prove the converse inclusion, starting by showing that for a string $y \in D' \cap L(X : Y)$ with $X, Y \in N$, there is a string x such that $x \in D$, $X \Longrightarrow_G^* xY$, and $h_1(x) = h_2(y)$. We omit the proof here.

The inclusion $h_1(D \cap L(G)) \supseteq h_2(D' \cap L(S : F))$ can be proved by considering the case where $X = S$ and $Y = F$ in the above claim. □

Since the class of context-free languages is closed under intersection with regular languages and morphisms, $H(Dyck \cap LOC(3)) \subseteq CF$ holds from the definition of Dyck language. Further, from Chomsky-Schützenberger characterization $CF = H(Dyck \cap REG)$, we have the following theorem.

Theorem 5. $CF = H(Dyck \cap LOC(3))$.

From the definition of Dyck language, we can easily show that for any Dyck language D, there is an insertion system γ of weight $(2, 0)$ which satisfies that $D = L(\gamma)$. Therefore, the next result follows from the fact that, for any i with $i \geq 2$, $Dyck \subset INS_i^0 \subset CF$ and Theorem 5.

Corollary 2. $CF = H(INS_i^0 \cap LOC(3))$ ($i \geq 2$).

Furthermore, from Theorem 1, we have the following corollary.

Corollary 3. $CF = H(INS_i^0 \cap LOC(k))$ ($i \geq 2$, $k \geq 3$).

4 Conclusion

In this paper, we have contributed to the study of insertion systems with new characterizations of context-free languages. Specifically, we have shown that $CF = H(INS_i^0 \cap LOC(k))$ $(i \geq 2, k \geq 3)$.

The following characterizations of regular languages in terms of insertion languages and strictly locally testable languages have shown in [8].

- $H(INS_1^0 \cap LOC(1)) \subset REG$.
- $REG = H(INS_1^0 \cap LOC(k))$ ($k \geq 2$).
- REG and $H(INS_i^0 \cap LOC(1))$ are incomparable ($i \geq 2$).
- $REG \subset H(INS_i^0 \cap LOC(k))$ ($i \geq 2, k \geq 2$).

The followings are open problems:
Can CF be represented as $CF = H(INS_i^0 \cap LOC(2))$ for $i \geq 2$?
Can CF be represented as $CF = H(INS_i^j \cap LOC(k))$ for $i, j \geq 1$ and $k \geq 1$?

Acknowledgements

The author is deeply indebted to T.Yokomori for his helpful discussions. This work is supported in part by Grant-in-Aid for the Research Institute for Science and Technology of Tokyo Denki University with no.Q07J-05.

References

1. Yokomori, T., Kobayashi, S.: Learning local languages and their application to DNA sequence analysis. IEEE Trans. Pattern Anal. Mach. Intell. **20**(10) (1998) 1067–1079
2. Head, T.: Splicing representations of strictly locally testable languages. Discrete Applied Math **87** (1998) 87–139
3. Păun, G., Pérez-Jiménez, M.J., Yokomori, T.: Representations and characterizations of languages in Chomsky hierarchy by means of insertion-deletion systems. Int. J. Found. Comput. Sci. **19**(4) (2008) 859–871
4. Păun, G., Rozenberg, G., Salomaa, A.: DNA Computing. New Computing Paradigms. Springer (1998)
5. Rozenberg, G., Salomaa, A., eds.: Handbook of formal languages. Springer-Verlag New York, Inc., New York, NY, USA (1997)
6. McNaughton, R., Papert, S.A.: Counter-Free Automata (M.I.T. research monograph no. 65). The MIT Press (1971)
7. Verlan, S.: On minimal context-free insertion-deletion systems. J. Autom. Lang. Comb. **12**(1) (2007) 317–328
8. Onodera, K.: New morphic characterizations of languages in chomsky hierarchy using insertion and locality. To appear in Proceedings of the 3rd International Conference on Language and Automata Theory and Applications (LATA 2009) (April 2-8, 2009, Tarragona, Spain)